ECHO - A Message-Based Framework for Metadata and Service Management

Keith Wichmann
Global Science and Technology, Inc.
6411 Ivy Lane, Suite 300
Greenbelt, MD 20770
wichmann@gst.com

Robin Pfister
Code 423
NASA/Goddard Space Flight Center
Greenbelt, MD 20771
robin.pfister@gsfc.nasa.gov

[*] ***Abstract*- ECHO (The Earth Observing System (EOS) ClearingHOuse) is being developed to provide flexibility to NASA's EOS Data and Information System to better meet the needs of the science community. This flexibility includes providing APIs for alternate user interfaces to support special needs in data access, providing APIs for brokering data services so specialized data services can be shared across the user community, and providing APIs for easy participation by a broad data provider community. Another major goal of ECHO is to support new data access paradigms that are not supported by today's EOSDIS architecture. To optimize performance and flexibility, ECHO was constructed to be a cache of metadata that describes EOS and related data and services. It uses eXtensible Markup Language (XML) as its primary method of exchanging information. Earth Science data providers can register with the system and provide copies of their metadata through an XML file. The system provides an XML message based interaction that allows clients to perform spatial, temporal and keyword searches for data sets or collections (directory) and data items or granules (inventory) contained in datasets. ECHO also acts as a broker, allowing a client to create a single order for items that span multiple providers. This order is then broken into pieces and handled individually, dealing with the asynchronous issues of communicating with providers. Additionally, ECHO will be adding the capability to broker or otherwise handle services on data, and the capability to extend the catalog service to support third-party search extensions. The taxonomy of services within ECHO's context is presented. The construction of ECHO was done using an abstract, layered approach that has wider applicability. This paper will show the layers of abstraction, how they are instantiated with the Earth Science application in mind, and how the framework that supports ECHO could be used in other contexts.**

I. INTRODUCTION

With the great quantities of data being generated by NASA's Earth Observing System and previous Earth Science missions, it is becoming more important to focus on providing a mechanism to allow people to find out what data is available in a single place. Rather than trying to centralize the control of the disparate types of data, NASA is pursuing an approach of creating a cache of the metadata in one place much like the popular web search engines do today. Since much of the data is kept on tape and must be staged and is frequently sent to the end user on some physical media, the ability to create a single order that deals with the differences of ordering data from different providers is also desirable. As users develop applications and utilities that act on this Earth Science data, it is also appropriate for there to be a mechanism for a service provider to be able to offer value-added functions on top of the data through this system. The opportunities presented by having NASA's Earth Science metadata in one place go beyond just providing flexibility for accessing data, it is also expected that new functions and services not currently envisioned will be conceived. The ability to augment the capabilities of the system without impacting current client users is crucial to this approach.

The system is envisioned to be an enabling system; such that other groups can have full and open access to its holdings (with allowances made for protected metadata) allowing them to build new human interfaces or even developing automated machine-based interfaces to the system. The system also holds as a guiding principle that the information technology industry should lead, and that the system shall adapt that technology to the purposes at hand. It is also anticipated that the ultimate users of such a system would have new features of their own to contribute. So designing for change is identified as a worthwhile goal. Global Science and Technology has built ECHO (Earth observing system ClearingHOuse) as an answer to these problems in the Earth Science domain. The architects of the system realized that these problems were shared in many domains, and created a layered architecture that will allow NASA to customize the system to many uses. That architecture is discussed in this paper, with some thoughts as to how the system could be reused.

While ECHO is seen as an enabler for clients that search metadata and create orders for the represented data, it is acknowledged that not every function should be provided by ECHO. For this reason, ECHO includes the concept of a search service. These services extend ECHO's search capabilities in some way that assists clients in their interactions with ECHO. These services allow ECHO to leverage capabilities provided by other systems without trying to replicate them. Additionally, ECHO is being augmented to allow providers of data services (those services that produced Earth

[*] The authors of this paper would like to acknowledge NASA's ESDIS project for supporting this work, as well as all the other people who have made this work possible and contributed directly to it. ESTO funding is directed at examining search services in ECHO.

Science data) to enter information into ECHO such that those services can be found and then applied to data that is (or might be) represented in ECHO.

A major feature of ECHO is that all interactions with it occur using XML as the base message format. This gives the system an extensible base upon which to build. The system uses a layered architecture combined with some code generation techniques to provide a structure that allows for rapid introduction and updating of business logic.

II. ECHO'S LAYERS

At its simplest, ECHO is a system that understands how to receive a message, perform some action as a result, and respond to the message with a result message. However, the result message may be the result of a very simple or a very complex action. As one drills deeper into ECHO's layered architecture, the core is found to be a database that can be used to store the clearinghouse of information. That clearinghouse can be searched using traditional text based methods that web search engines use, but also enables temporal and spatial types of searches that are appropriate for the geospatial data referenced here. The data model for ECHO is currently based on the EOS data model. Each layer of the architecture and its corresponding responsibilities in the system, is outlined below.

A. Session Management

In any instantiation of the ECHO framework, the outermost layer with which client applications communicate is the Session Manager (SM). This layer understands XML messages and how to validate them, understands that there are services which may be invoked, and understands that users have types that can be used to determine which services they can access. The SM has no idea what the services do. The client opens up a connection via Java Remote Method Invocation (RMI) and transmits an XML message to the SM, and the SM replies to that message with an XML response message.

The SM is implemented as a Session Bean in the J2EE environment. It uses a validating parser to accept the XML message and validate it against its Document Type Definition (DTD). The message is then turned into the appropriate set of Java objects that represent the request. In this way, no other part of the system needs to understand how to parse an XML message (unless there are embedded XML messages contained within the message, as is the case with a query). The conversion is enabled by a mapping file, so changes in the API do not require changes in the SM, only an update to the mapping file. The SM looks at the first tag in the XML message to determine which service needs to be invoked. If the XML tag does not match the known services, then the SM responds with an error message. The message is validated against its DTD before it ever reaches any of the business logic of the system. The SM currently accepts RMI connections, but converters allow the system to accept other methods such as Simple Object Access Protocol (SOAP). In essence, the SM has three methods: Login, Logout and Perform. The Perform method simply accepts an XML message and returns one.

B. Service Management

Transactions are grouped together as Services. Example services in the current version of ECHO include User Account Service, Registration Service, Order Entry Service, Catalog Service and Provide Order Management Service. The Service can be configured to control access to its transactions such that only users of a certain type can use that service. Currently, there are three types of users in the system: guests, registered users, and registered providers.

The Service is aware of what transactions are available. The second tag in the message contains the name of the transaction that is being requested. This is translated into a Java method invocation that will perform the transaction. This is also the first level that the online documentation becomes a factor. Each service has a description that is used in combination with the list of transactions as the high level documentation of that service. This documentation is hyperlinked to provide increasing amounts of detail. This is automatically generated by the Synchronicity Generator, a component of the framework that supports keeping all aspects of the API related development items in synch.

The remainder of the transaction's message is used to describe any parameters that are to be passed into the system. Some transactions which are particularly flexible require that a string be passed into the transaction as a payload which can then be separately interpreted. Queries are an example of this in that a query string is passed as an embedded part of the query. In the current system, that query is expressed as XML. Therefore, the system supports passing an XML message that has an embedded XML message which is separately parsed, validated, interpreted and acted upon.

C. Business Logic

This is the layer of the architecture where the particular logic that the system needs is implemented. This logic can be simple create, remove, update and delete transactions on persisted objects, or it can be as complicated as queries on a complex catalog of represented data or state-aware functions. The framework architecture is designed to provide the interface to the business logic in a configurable, easily augmented fashion. Therefore, when new functionality is added to the system, it is a matter of some simple configuration and the introduction of the new business logic itself.

D. Query Management

The basic goal for ECHO is to provide search and order capabilities on Earth Science data that is represented by a very complex metadata model and is supplied by multiple data providers. This necessitates a method for searching that

metadata and interacting with potentially very large result sets. It is also desirable to allow for multiple query languages. The Open GIS Consortium model is used as a basis. This supports submitting a query and specifying the language of the query, as well as asking for different types of results.

Since the ECHO system is designed to enable other systems, but there is more metadata stored in the system than is practical to transmit to clients for every hit, it is very important to provide transactions that give the client enough flexibility to manage their interactions in a profitable fashion. First, a query is sent to the system. This transaction will return once the query has completed. Once a query has completed, the database maintains a result set with the identifiers for each item that matched the query. The client then has a mechanism to segment those results both vertically and horizontally. First, there is a paging mechanism that allows the client to request results starting at some offset and for some extent from that offset. In this way, a client can request the first 10 results, the second 10, and so on similar to how web search engines function. Second, the client can request all of the metadata be returned for each hit, or only a subset of it. This allows a client to only retrieve metadata that it needs, or to allow iterative "drilling down" into the metadata as a particular hit is perceived to be of interest.

The rendering of results in the message can also be configured through the Present transaction. The rendering can be any text based format that the business logic supports. In the Earth Science domain, the metadata represented has geospatial and temporal characteristics. Therefore, the system must understand how to search this type of data in addition to the more traditional types. Furthermore, results can be saved for later interactions, and queries can also be saved. This further supports enabling clients to provide a wide variety of services to their customers. Evaluation of whether a user has access to particular metadata occurs at the results presentation level, minimizing the impact on performance.

In ECHO, the catalog schema is rather complex, consisting of about 50 tables. The query language for interrogating the metadata holdings is very general by nature, and it is important to be able to minimize performance impacts on the system from user queries. ECHO contains a mechanism for minimizing the number of joins needed.

E. Data Model

All of the data within the system is maintained in a relational database. Both persisted objects with which transactions interact directly, and the more complicated schema of the catalog are stored here. An object-relational mapping tool that is part of the J2EE environment is used to help persist objects in the database. The catalog is treated as a database schema of its own, and JDBC is used to interact with it. The remainder of the framework, with the exception of the catalog service, is ignorant of the content of the catalog and can therefore be reused in any number of situations. The Catalog Service already has support for geospatial and temporal searches which could optionally be used in other situations as well.

III. SERVICES

There are various service concepts used in ECHO. There are Clearinghouse Services provided by ECHO. These are internal services that support the clearinghouse functions. Then, through APIs, ECHO supports various types of external services that are made available by service providers and shared with users of ECHO. These include Search Services, Data and Science Services, and in the future, Administrative Services. These are described in the following paragraphs.

A. Clearinghouse Services

Based on the framework, ECHO provides a variety of services that directly support its mission. The Catalog Service is the mechanism used to query and retrieve results from the system. It supports multiple query languages and result formats. It supports the chunking of result sets so that large result sets can be effectively handled by clients. It also supports saving and restoring both queries and result sets by registered users of the system for later use. The Order Entry Service allows both guest users of the system and registered users to build, validate, quote and place orders, as well as to examine the orders that they have created. The Registration Service allows a guest to create a registered user account for themselves, as well as allows prospective providers to alert the operational staff of their interest. The User Account Service allows a registered user to maintain address, email and phone information so that a client may use that to fill in information for orders. It also supports checking a user's order history. The Provider Account Service provides a similar interface for ECHO providers to describe themselves and check their order history. The Group Management Service allows the creation and management of named groups of users that can then be used to grant special metadata or order access privileges to. The Data Management Service is used to create rules that describe what metadata in the system is protected, and who is given access to it. It supports restricting and granting access based on individual granules, collections, or sets of granules within a collection based on temporal conditions or a quality flag. Finally, the Provider Order Management Service allows a provider to give ECHO information about the status of an order, including the ability to cancel an order.

B. Search Services

This concept is in the early stages of development, but is foreseen to be an excellent infusion point for other NASA Earth Science search techniques. The concept is to extend the as-built search capabilities of ECHO with additional information that improves the end-user experience (or perhaps the client developer's experience). Several existing and past ESTO prototypes fit this description. Perhaps the best example is that of the Gazetteer. This system maintains a mapping

between place names and their locations on the planet. It is reasonable to think that a user (or client developer) would want to search for geospatial data by using the name of the place they are interested in. For instance, a user may want to search for data that matches Maryland. While some states lend themselves to bounding box representations very well, Maryland is not one of them. The polygon that represents Maryland could be maintained at a Gazetteer, and the Gazetteer could offer a search service through ECHO that converts the place names which it understands into polygons which ECHO understands. In this way, ECHO's capabilities are augmented in a way that benefits clients, but ECHO is not burdened with the added responsibility of maintaining political boundaries.

Similarly, other search services can be envisioned that would assist the client's job of providing an effective interface to the user. A Thesaurus could be used to map from one community's set of keywords to the common set of keywords that ECHO provides. Similar to the Gazetteer, the mapping would be applied to a search query before it was executed by ECHO, transforming keywords that would have produced undesirable results into ones that produce the desired results. Also, a mapping could be performed on the result of the search such that the common keywords of ECHO are converted into the community specific keywords that best serve those users. There may be many instances of Thesauri, each with its own target community, but all using the same search service mechanism.

One final example would be Coincidence Search. In this case, the client's aim is to find a set of data in which multiple observations from different instruments of a single area within a tolerance of a given window of time. In other words, a user may want to find all the places in North America where a MISR and a TRMM observation were taken within 5 minutes of each other. The coincidence can be calculated and translated into a query to ECHO which produces results that are physically rather than theoretically available.

C. Data and Science Services

Another extension point under development for ECHO is the ability to allow ECHO clients to find out about and use third-party provided services. Obviously, these services are in the Earth Science domain. ECHO refers to them as data and science services because the end result of their execution is that some piece of Earth Science data is generated. Service outputs can mimic types of data that already exist in the archive, or they can generate new information that is not archived. ECHO divides this type of service into four subtypes based on how the client and service interact with ECHO.

Some services require no interaction with ECHO and simply present a web page that lets a user know everything they need. In this case, ECHO facilitates finding out about the service, and takes no further action. These are referred to as Advertised Data Services.

Other services add the need for ECHO to associate which data represented in ECHO will be used for the invocation (binding) of the service. The client continues to invoke the service directly to the service provider, but ECHO provides an additional service of pre-populating a subset of the service parameters. In this way, various types of services that are invoked via a URI (Web Mapping Service, DODS, etc.) can be invoked by a client in the same fashion without necessarily having to understand all aspects of the standard for putting together the URI. ECHO abstracts that problem, and refers to these as Context-Based Data Services.

Services that require being ordered because of payment involved or some time delay that will be involved in invoking the service can be brokered through ECHO. ECHO will bind to the service upon request from the user for that service. Future implementations of ECHO might support the chaining of brokered services to produce new aggregate services. These are referred to as Brokered Data Services. One special type of brokered service is those that are offered by a data provider and are ordered as an option when the data is ordered. These are referred to as Order Option Data Services.

D. Administrative Services

In the future, other externally provided services will also be supported through ECHO. An example is a billing and accounting service that might be provided by a single service provider but might serve several data and service providers to allow them to charge for their data and/or service.

IV. CONCLUSION

The framework described here, of which ECHO is the first instantiation, is useful in many applications. At the highest level, ECHO is a system for receiving messages and responding to them. The framework supports the definition of the messages and the simple plugging in of business logic that is triggered by the message. At the next level, ECHO is a catalog aware message system. So, in addition to other messages, there is the ability to send queries to the system which are in turn evaluated and the results made available for interaction. There is also the capability to augment these searches with geospatial and temporal parameters. Finally, there may be domains in which the Earth Science data model could be used as a basis. If not, the framework allows for a substitution of the catalog to accommodate a wide variety of needs.

With the advent of the web services movement, it will be interesting to see how well this framework fits into it. One new task that ECHO will have to address is the introduction of Earth Science data services into the system. Web services are being examined as being an appropriate mechanism for invoking these services, so ECHO will be moving in this direction in the near future. Search services will allow the augmentation of ECHO's search capabilities in a componentized fashion, allowing for other projects to infuse themselves into a system with operational data.